ARMY RESEARCH LABORATORY

A MATLAB Radar Range Equation and Probability of Detection Evaluation Tool

Barry Scheiner

ARL-TR-1806

January 1999

Approved for public release; distribution unlimited.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-1806 **January 1999**

A MATLAB Radar Range Equation and Probability of Detection Evaluation Tool

Barry Scheiner
Sensors and Electron Devices Directorate

Abstract

I have developed a graphical user interface (GUI) that makes the process of evaluating the radar range equation faster and more convenient. From this GUI, one can enter the input parameters necessary to compute the signal-to-noise ratio (SNR) and probability of detection (P_d) of a radar system as a function of range, and view the results graphically. The GUI and the SNR and P_d computations are coded using the MATLAB programming language. I review the derivation of the equation used to evaluate the P_d of a signal detected by a linear quadrature detector and provide examples of the GUI's operation.

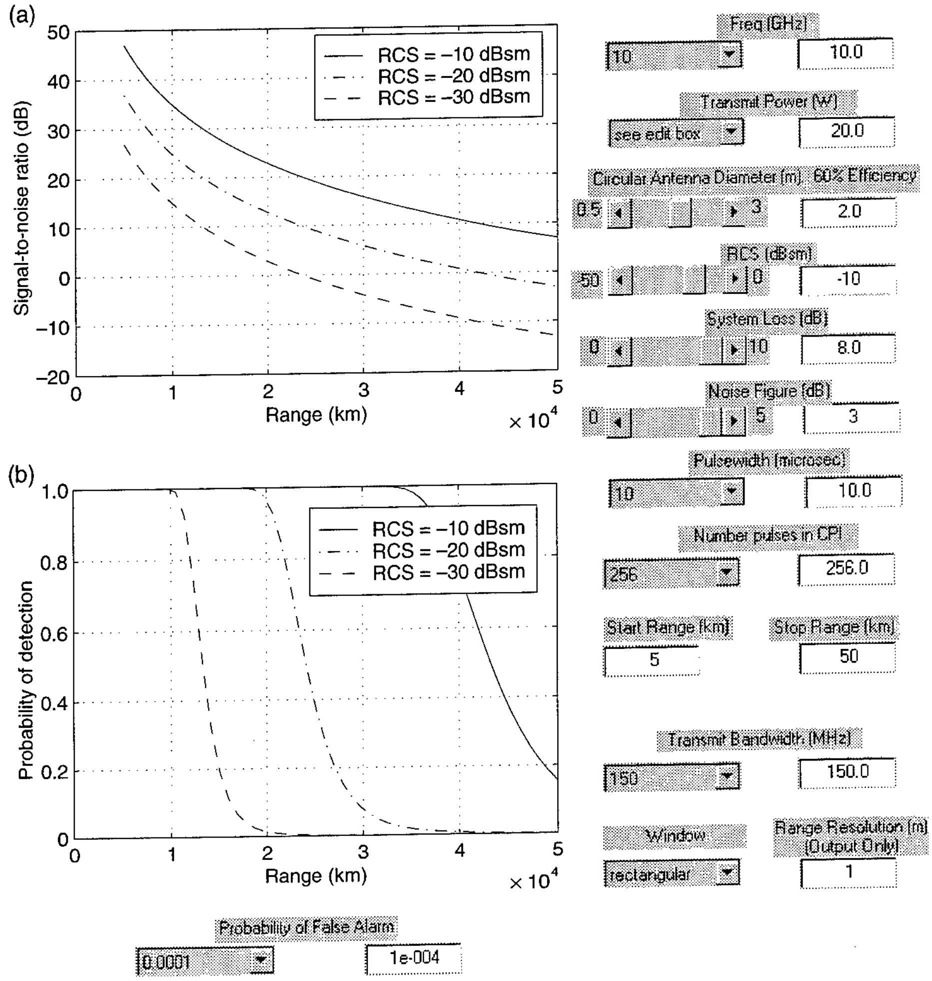
Contents

Figures

1. Introduction

Evaluating the radar range equation and obtaining probability of detection (P_d) as a function of range are frequently useful when performing a radar systems analysis. A MATLAB graphical user interface (GUI) has been developed to do this conveniently and flexibly. Shown in figure 1, the GUI allows parameters to be entered from either the pop-up boxes, sliders, or editable text boxes. The values taken from the text boxes are used to calculate curves of signal-to-noise ratio (SNR) versus range from the radar range equation (fig. 1 (a)), and these curves are plotted as a function of range for three values of the radar cross section (RCS). A second plot of P_d (fig. 1 (b)) is generated based on the computed SNR values and a probability of false alarm (P_{fa}) entry. This evaluation tool, consisting of a MATLAB GUI and MATLAB code for computation, is intended to be used as a first cut for system design, since it does not have the level of detail to support an advanced study. Although it is flexible enough to support most surveillance radar designs, such as the Army Research Laboratory's (ARL's) low-cost enabling radar technology (LCERT) radar or the Yuma Proving Ground (YPG) instrumentation radar programs, it would have to be modified to address other types of radars. However, it would not be difficult to extend or modify the capabilities of this tool or to add features, such as a detailed system-loss budget including atmospheric attenuation or antenna shapes other than circular. Modifications could also be made to this MATLAB code to change the parameter computations or to read out intermediate results.

Figure 1. YPG point design results: (a) signal to noise versus range and (b) probability of detection versus range.

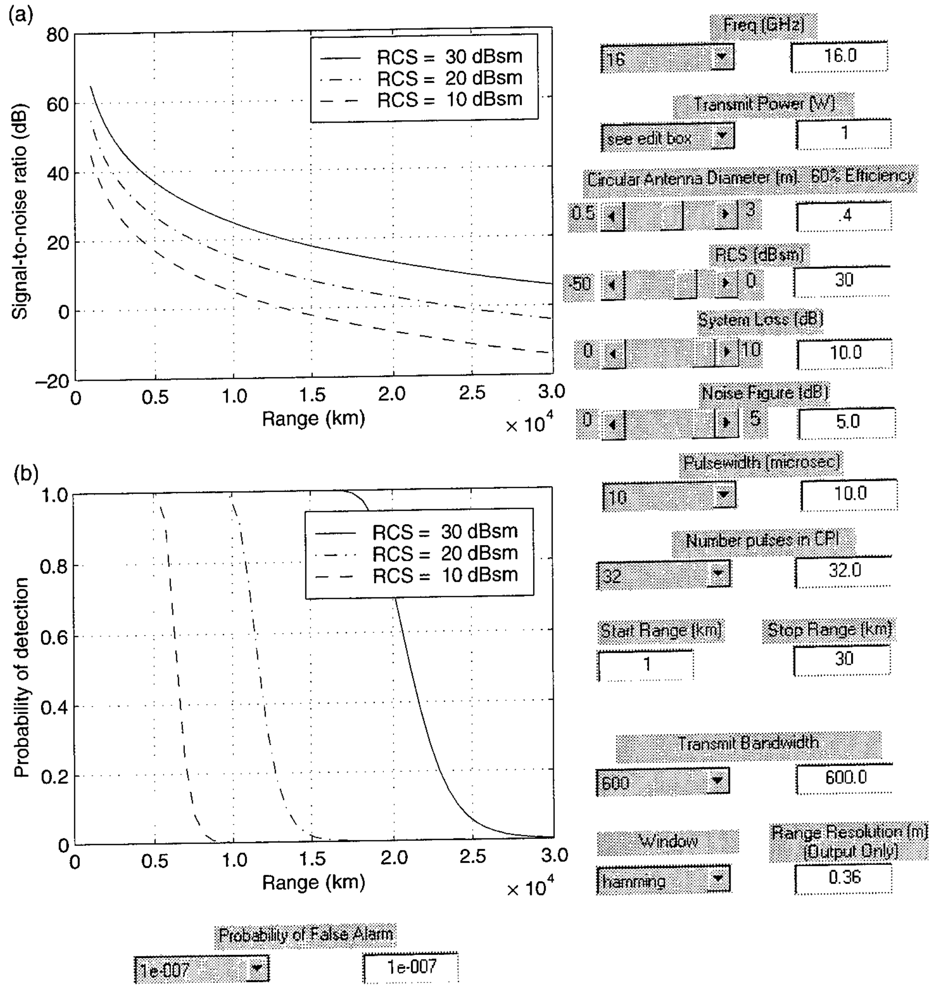


2. Graphical User Interface

The GUI is called with the command "rrefig" at the MATLAB prompt. The command "help rrefig" will print a synopsis of how to use the GUI. To use the GUI, enter the desired parameter values into either the sliders, pop-up boxes, or editable text boxes. After each entry, a new set of SNR and P_d curves is computed. Since this computation takes about 3 s (almost entirely because of the P_d computation), the response of the sliders may be somewhat slow. The values entered into the pop-ups or sliders are copied to the associated editable text boxes. Conversely, a value entered into an editable text box causes the associated pop-up to read "see edit box." However, a value entered into an editable text box does not cause any change in the position of the associated slider. The transmitted bandwidth and window function inputs are not used as inputs to the radar range equation. Rather, they are used to compute the theoretical range resolution that would be obtained. The example computation shown in figure 1 is based on a conceptual design for a projectile tracking instrumentation radar. The RCS values used are based on a range of generic artillery projectiles. Another example, shown in figure 2, is based on the LCERT system design.*

*Eric Adler et al, "Low-Cost Enabling Technology for Multimode Radar Requirements," *Record of the IEEE 1998 Radar Conference*, 12–13 May 1998.

Figure 2. LCERT field test parameters: (a) signal to noise versus range and (b) probability of detection versus range.



3. Radar Range Equation

One can use a form of the radar range equation found in Skolnik's handbook* to obtain SNR as a function of range and RCS:

$$\mathrm{SNR} = \frac{P_t G_A{}^2 \sigma \lambda^2 G_{ibw} G_{dop}}{(4\pi)^3 kTBR^4 LN_f}\ , \tag{1}$$

where

P_t = peak transmitted power,

G_A = antenna gain,

σ = radar cross section of target,

λ = wavelength of transmitted carrier,

G_{ibw} = time-bandwidth product gain,

G_{dop} = Doppler processing gain,

k = Boltzman's constant,

T = ambient temperature,

B = modulation bandwidth of transmitted signal,

R = range to target,

L = system losses, and

N_f = noise figure.

Although the equation is exact, the parameters used to obtain the SNR are often approximations or based on engineering judgment. Obviously, the user must apply knowledge of the problem and experience to obtain a reasonable result, and the value of this tool is that it makes this process more efficient. The details of the SNR computation may be found in the comments in the MATLAB code shown in the appendix of this report. The antenna gain is computed based on entering the diameter of a circular antenna with an efficiency of 60 percent.

*M. Skolnik, *Radar Handbook*, 2d edition, 1990, p 2.6.

4. Probability of Detection

In this section, I will discuss the derivation of the equation that I used to compute the P_d as a function of SNR for a given P_{fa}. The derivations are based on course notes by Trunk* and a book by McDonough and Whalen.† I will consider the situation where an input signal $v(t)$ is processed through a linear quadrature detector, as shown in figure 3. The integral function in the quadrature detector represents a low pass filter. The output of the quadrature detector is

$$q = \left[\left(\int_0^\tau v(t) \sin(\omega_c t)\, dt \right)^2 + \left(\int_0^\tau v(t) \cos(\omega_c t)\, dt \right)^2 \right]^{1/2} , \tag{2}$$

where τ = measurement period, $0 \le t \le \tau$, and ω_c = carrier frequency.

One can now postulate two possibilities for $v(t)$. It may consist only of white, Gaussian, zero-mean noise, or it may consist of a sinusoidal signal plus the noise. Thus I define two hypotheses:

$$\begin{aligned} H_0&: \quad v(t) = n(t) \\ H_1&: \quad v(t) = n(t) + A \sin(\omega_c t + \theta) , \end{aligned} \tag{3}$$

where $n(t) \sim G(0, \sigma_n^2)$,

ω_c = carrier frequency,

A = amplitude (peak-to-peak),

θ = random receive phase,

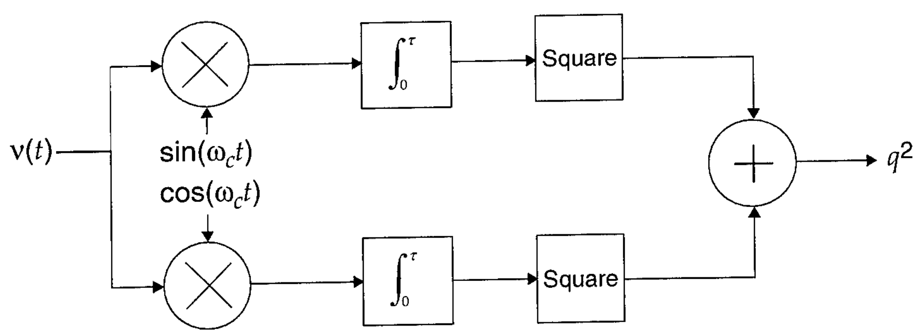
σ_n^2 = variance of noise signal,

$G(\ )$ = Gaussian (normal) distribution,

and the signal-to-noise ratio of power is defined as

$$\mathrm{SNR} = \frac{A^2}{2\sigma_n^2} . \tag{4}$$

Figure 3. Linear detector.



*G. V. Trunk, *Detection and Estimation Theory Handbook*, notes for Johns Hopkins course 525.728, p 60.

†R. N. McDonough and A. D. Whalen, *Detection of Signals in Noise*, 2d edition, Academic Press, 1995, pp 159–166.

(In eq (4), I define the SNR as power rather than as energy as in McDonough's definition (eq (7.32) and (7.54)). If τ is normalized to equal 1 in equation (7.32), then the power for a complex signal is

$$P = 2\left[\left(\frac{A}{2\sqrt{2}}\right)^2 + \left(\frac{A}{2\sqrt{2}}\right)^2\right] = \frac{A^2}{2} \ , \tag{5}$$

where the factor of 2 outside the brackets is introduced for convenience, as it was in equation (7.25) of McDonough's work.

The Neyman-Pearson criterion* is typically used in radar detection problems to determine an expression for P_d for a given P_{fa}. It specifies a procedure to test one hypothesis against another and is a most powerful test—P_d is maximized for a P_{fa} less than or equal to some value. To simplify the derivation, one can first assume continuous probability density functions (pdf's) for the distributions of interest. Then one can assume that the envelope of the input signal does not vary during the measurement period τ. If it had, one could still show that the test is uniformly most powerful for the amplitude and could proceed the same way. One could then compute the P_d for a target with Rayleigh fading, or for a Swerling II target when multiple coherent processing intervals (cpi's) are integrated noncoherently. Finally, one assumes that although the phase angle θ of the input signal is unknown, it has a uniform random pdf. Although the random variable in the signal makes H_1 a composite hypothesis, it can be integrated out, with no change in the form of the pdf for H_1, and now H_1 becomes a simple hypothesis.

One then defines the P_{fa} as the probability that a detection D_1 is declared when hypthesis H_0 in equation (3) is correct:

$$P_{fa} = P(D_1 \mid H_0) = \int_T^{\infty} p_0(q)dq \ , \tag{6}$$

where T is defined as the detection threshold level and $p_0(q)$ is the pdf associated with hypothesis H_0. Similarly, the P_d is defined as the probability that a detection D_1 is declared when the hypothesis H_1 in equation (3) is correct:

$$P_d = P(D_1 \mid H_1) = \int_T^{\infty} p_1(q)dq \ , \tag{7}$$

where $p_1(q)$ is the pdf associated with hypothesis H_1. The threshold T will be some function of q and is found by using the likelihood ratio test. This test states that you decide H_1 if

$$\lambda(q) = \frac{p_1(q)}{p_0(q)} \geq T \ . \tag{8}$$

*R. N. McDonough and A. D. Whalen, *Detection of Signals in Noise*, 2d edition, Academic Press, 1995, pp 159–166.

To find the ratio in equation (8), one must determine the numerator and denominator pdf's. McDonough* derives the pdf of a linearly detected narrowband signal plus narrowband noise. It is known as the Rician density function, and for this problem is given by

$$p_1(q) = \left(\frac{q}{\sigma_n^2}\right)\exp\left[\frac{-(q^2 + A^2)}{2\sigma_n^2}\right] I_0\left(\frac{Aq}{\sigma_n^2}\right) , \tag{9}$$

where I_0 = Bessel function of the first kind order zero. If the signal q is equal to zero, then $I_0(0) = 1$, and the resulting density will be that of integrated noise, which is the Rayleigh probability density given by

$$p_0(q) = \left(\frac{q}{\sigma_n^2}\right)\exp\left[\frac{-q^2}{2\sigma_n^2}\right] . \tag{10}$$

The threshold can now be found by substituting equations (9) and (10) into equation (8):

$$\lambda(q) = \frac{p_1(q)}{p_0(q)} = \frac{\left(\frac{q}{\sigma_n^2}\right)\exp\left[\frac{-(q^2 + A^2)}{2\sigma_n^2}\right] I_0\left(\frac{Aq}{\sigma_n^2}\right)}{\left(\frac{q}{\sigma_n^2}\right)\exp\left[\frac{-q^2}{2\sigma_n^2}\right]} = \exp\left[\frac{-A^2}{2\sigma_n^2}\right] I_0\left(\frac{Aq}{\sigma_n^2}\right) \geq T . \tag{11}$$

In equation (11), only the term inside the argument of the Bessel function contains the term q. So the threshold test reduces to

$$I_0\left(\frac{Aq}{\sigma_n^2}\right) \geq T , \tag{12}$$

where constants have been absorbed into q_t. Since the Bessel function is a monotonically increasing function of q, the ratio test can be performed as a linear function of q so that

$$q \geq T . \tag{13}$$

From equation 13, one can set $T = q_t$ and use equations (6) and (10) to find q_t in terms of a given value of P_{fa}:

$$P_{fa} = \int_{q_t}^{\infty} \left(\frac{q}{\sigma_n^2}\right)\exp\left[\frac{-q^2}{2\sigma_n^2}\right] dq = \exp\left[\frac{-q_t^2}{2\sigma_n^2}\right] . \tag{14}$$

By rearranging, one obtains

$$\frac{q_t}{\sigma} = [-2\ln(P_{fa})]^{1/2} . \tag{15}$$

---

*R. N. McDonough and A. D. Whalen, *Detection of Signals in Noise*, 2d edition, Academic Press, 1995, pp 129–130.

Substituting equation (9) into equation (7), the P_d is

$$P_d = \int_{q_t}^{\infty} \left(\frac{q}{\sigma_n^2}\right) \exp\left[\frac{-\left(q^2 + A^2\right)}{2\sigma_n^2}\right] I_0\left(\frac{Aq}{\sigma_n^2}\right) dq \ . \tag{16}$$

The substitution, $v = q/\sigma_n^2$, is now made in equation (16) to put the integral in the form of the Marcum Q function,* which is more convenient for calculating. Also, by using the expression for SNR in equation (4), equation (16) becomes

$$Q\left(\sqrt{\mathrm{SNR}}, \frac{q_t}{\sigma}\right) = 1 - \int_0^{q_t/\sigma} v \exp\left[\frac{-\left(v^2 + \mathrm{SNR}\right)}{2}\right] I_0\left(\sqrt{\mathrm{SNR}}\ v\right) dv \ . \tag{17}$$

This is the expression used to compute the P_d for a given P_{fa}, where q_t/σ is found from equation (15). The code used to evaluate equation (17) in terms of equation (15) is found in the appendix. It is important to note that the SNR as defined in equation (4) is lower than the definition in Skolnik† by a factor of 2 or 3 dB. The Skolnik definition is used in the MATLAB GUI so that the SNR calculated from the radar range equation matches his curves in figure 2.3 of his publication rather than McDonough's curves in figure 7.3 of his publication. Also note that the SNR as defined above is in units of volts per volt, not decibels. In equation (1), the computation is performed in decibels. So one converts to the form used in equation (17):

$$\sqrt{\mathrm{SNR}} = 10 \log_{10}\left(\frac{\mathrm{SNR_{dB}}}{20}\right) \ . \tag{18}$$

---

*R. N. McDonough and A. D. Whalen, *Detection of Signals in Noise*, 2d edition, Academic Press, 1995, p 131.

†M. Skolnik, *Radar Handbook*, 2d edition, 1990, p 2.6.

5. Conclusion

A MATLAB GUI has been developed that can be used to quickly and easily determine the SNR and P_d as a function of the radar range equation parameters and the P_{fa}. I have used it to plot curves of SNR and P_d versus range for two different radar examples: the YPG instrumentation radar and the LCERT. The equation used to compute P_d as a function of SNR and P_{fa} was derived using the Neyman-Pearson criterion and procedure. One could further develop the capabilities of this GUI by adding new functionality and features.

Appendix. Code for Radar Range Equation and P_d GUI

```
%            FILE: rre
%            DATE: 7/19/98
%
% This function is the target of callbacks in the rrefig GUI. Radar range equation
% parameters and the Pfa are read from the editable text windows, and used to
% compute SNR and Pd as a function of SNR for the given Pfa. The vulues from the
% popups and sliders are read into the associated editable text boxes. After
% reading the parameters, the SNR as a function of range is computed for 3 RCS
% values. Then, the Pd as a function of range is computed based on the SNR values
% from the radar range equation. See technical report "A MALAB Radar Range Equation
% and Probability of Detection Evaluation Tool".
% The MATLAB copy figure function does not copy the popups, sliders, or boxes.
% To copy everything in the GUI to a eps file with a tiff preview feature, use
% command "print -depsc2 -tiff <path.filename>".
%
%   trans_pwr          % Transmitter power in Watts.
%   antenna_diam       % Antenna diameter in meters
%   rf_loss            % RF system hardware loss
%   frequency          % Transmitter frequency in GHz (1 GHz = 1)
%   rcs_start          % RCS in dBsm
%   noise_figure       % Receiver LNA noise figure in dB
%   pulsewidth         % Pulsewidth in microseconds.
%   cpi_pulses         % Number pulses in coherent processing interval
%  window              % Rectangular for no window, or Hamming
%
%   snr                % Power signal-to-noise ratio, in dB.

% action == 1 reserved for initial conditions if required.

function rre(action)

%************ READ PARAMETER VALUES FROM GUI ****************************

       % Execute this block from every action on the GUI callbacks. For popups
   % and sliders, their callbacks put their values into the editable text block
   % which is then read in this section.
if action ~= 1
   h = findobj('Tag', 'Freqe');          % Finds the object with the tag 'Freqe'.
   frequency = eval(get(h, 'String'));          %In this case, eval is like str2num.
             % frequency = eval(get(gcbo, 'String')) used in celsius example
               % Use code below to get value from popup; see p 3-22 of GUI manual
               %h = findobj('Tag', 'Freq');
             %value = get(h, 'Value');
               %string = get(h, 'String');
               %frequency = str2num(string{value});

      % If the action is from the freqe editable text block, set the value of the
      % freq popup to "see edit box". Similarly for the other popups.
   if action == 3
        h = findobj('Tag', 'Freq');
     set(h, 'Value', 5)          % Set the value of the popup, 'Freq', to 5, which is
                                        % "see edit box" in the string cell for 'Freq'
   end
```

```
    h = findobj('Tag', 'Powere');
    trans_pwr = eval(get(h, 'String'));
    if action == 4
            h = findobj('Tag', 'Power');
       set(h, 'Value', 6)
    end

    h = findobj('Tag', 'Diame');
    antenna_diam = eval(get(h, 'String'));

    h = findobj('Tag', 'RCSe');
    rcs_start = eval(get(h, 'String'));

    h = findobj('Tag', 'Losse');
    rf_loss = eval(get(h, 'String'));

    h = findobj('Tag', 'Nfe');
    noise_figure = eval(get(h, 'String'));

    h = findobj('Tag', 'Pwe');
    pulsewidth = eval(get(h, 'String'));
    if action == 5
            h = findobj('Tag', 'Pw');
       set(h, 'Value', 7)
    end

    h = findobj('Tag', 'Numbere');
    cpi_pulses = eval(get(h, 'String'));
    if action == 6
            h = findobj('Tag', 'Number');
       set(h, 'Value', 7)
    end

    h = findobj('Tag', 'Pfae');
    pfa = eval(get(h, 'String'));
    if action == 7
            h = findobj('Tag', 'Pfa');
       set(h, 'Value', 8)
    end

    h = findobj('Tag', 'Strt_rnge');
    start_range = eval(get(h, 'String')) * 1.e3;

    h = findobj('Tag', 'Stp_rnge');
    stop_range = eval(get(h, 'String')) * 1.e3;

    h = findobj('Tag', 'BWe');
    bandwidth = eval(get(h, 'String'));
    if action == 8
            h = findobj('Tag', 'BW');
       set(h, 'Value', 6)
    end

    h = findobj('Tag', 'Window');
    value = get(h, 'Value');
        string = get(h, 'String');
        window = string{value};

end
```

```
%****************** EVALUATE RADAR RANGE EQUATION ******************

c = 3e8;                                % Velocity of light (m/sec).
A = pi * antenna_diam ^2 / 4;
wavelength = c / (frequency * 1.e9);
antenna_gain = 10 * log10(.6 * 4 * pi * A / wavelength ^ 2);

four_pi = 10 * log10((4 * pi)^3);

pt = 10 * log10(trans_pwr);
lambda_sq = 2 * 10 * log10(c / (frequency * 1.e9));
ktb = 10 * log10(1.38e-23 * 300 *(bandwidth * 1.e6));
t_bw_gain = 10 * log10(pulsewidth * bandwidth);
dop_gain = 10 * log10(cpi_pulses);

range = linspace(start_range, stop_range, 100);
rcs(1) = rcs_start;
delta_rcs = -10;
for ii = 1 : 3                          % Loop over 3 RCS values
   snr(ii, :) = pt + lambda_sq + 2*antenna_gain + t_bw_gain + dop_gain + rcs(ii)...
     - four_pi - ktb - 40 * log10(range) - noise_figure - rf_loss;
       rcs(ii+1) = rcs(ii) + delta_rcs;
end

% **************** EVALUATE RANGE RESOLUTION **************************

res = c /(2 * (bandwidth * 1.e6));
if window == 'hamming    '
   res = res * 1.44;
end
h = findobj('Tag', 'Res');
set(h, 'String', res)


% **************** EVALUATE Pd FUNCTION ******************************
       % Taken from McDonough and Whalen, "Detection of Signals in Noise", 2nd ed
   % eqn 7.53
global alpha         % Make alpha global to pass it to marcum_q_fn.
beta = sqrt(-2*(log(pfa)));

%snr1 = 5 : 15;
range1 = linspace(start_range, stop_range, 50);
lr1 = length(range1);
for ii = 1 : 3
  for jj = 1 : lr1
    % In the Marcum Q function, the SNR is defined as a voltage ratio, but
     % the rre SNR is a power ratio. To take the square root, divide by 20 instead of
10.
  alpha = 10.^((snr(ii, 2 * jj) + 3) / 20);
    % Add 3dB to SNR to match Skolnik, p2.20
    % Don't add 3dB to SNR to match McDonough, p 261.
    % Care should be taken in the use of tolerance for the integration. Larger toler-
ances
    % will run faster, but will generate errors in Pd for small values of Pd. The
errors
    % are not noticeable in the linear scale plots.
   pd(ii, jj) = 1 - quad('marcum_q_fn', 0, beta, [1.e-1  1.e-3]);
  end
end
```

```
l1 = sprintf('RCS = %3.0f dBsm', rcs(1));  % Use these in the legends.
l2 = sprintf('RCS = %3.0f dBsm', rcs(2));
l3 = sprintf('RCS = %3.0f dBsm', rcs(3));

haxes = findobj('Tag', 'Axes1');
set(haxes, 'nextplot', 'replacechildren'); % These two lines keep plots from printing
                                           %  to outer figure box.
axes(haxes)
x_lims = get(haxes, 'xlim');      % Use this to make xaxis in second plot consistent
plot(range, snr(1, :), 'k-', range, snr(2, :), 'k-.', range, snr(3, :), 'k-')
grid on; title('Signal-to-Noise vs. Range')
xlabel('Range (km)');  ylabel('Signal to Noise Ratio (dB)')
legend(l1, l2, l3, 0)

haxes = findobj('Tag', 'Axes2');
set(haxes, 'nextplot', 'replacechildren');
axes(haxes)
plot(range1, pd(1, :), 'k-', range1, pd(2, :), 'k-.', range1, pd(3, :), 'k-')
axis([x_lims(1)  x_lims(2) 0 1])%        % Use x axis limits from first plot.
grid on; title('Probability of Detection (Pd) vs. Range (Skolnik p2.20)')
xlabel('Range (km)');  ylabel('Pd')
legend(l1, l2, l3)
```

```
%         FILE: marcum_q_fn.m
%         DATE: 7/29/98

% Use this function as integrand to evaluate the Marcum Q
% function that provides the probability of detection for
% a single pulse out of a quadrature detector. Called from
% rre.m which calculates the radar range equation.


function f = marcum_q_fn(v)

global alpha

f = v .* exp(-(v.^2 + alpha.^2)/2).* besseli(0, alpha.*v);
```

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Ofc of the Dir Rsrch and Engrg
Attn R Menz
Pentagon Rm 3E1089
Washington DC 20301-3080

Ofc of the Secy of Defns
Attn ODDRE (R&AT)
Attn ODDRE (R&AT) S Gontarek
The Pentagon
Washington DC 20301-7100

OSD
Attn OUSD(A&T)/ODDR&E(R) R J Trew
Washington DC 20301-7100

AMCOM MRDEC
Attn AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

CECOM
Attn PM GPS COL S Young
FT Monmouth NJ 07703

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Prsnnl
Attn J Hiller
The Pentagon Rm 2C733
Washington DC 20301-0300

Hdqtrs Dept of the Army
Attn DAMO-FDT D Schmidt
400 Army Pentagon Rm 3C514
Washington DC 20301-0460

US Army Armament Rsrch Dev & Engrg Ctr
Attn AMSTA-AR-TD M Fisette
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Edgewood RDEC
Attn SCBRD-TD G Resnick
Aberdeen Proving Ground MD 21010-5423

US Army Info Sys Engrg Cmnd
Attn ASQB-OTD F Jenia
FT Huachuca AZ 85613-5300

US Army Natick RDEC Acting Techl Dir
Attn SSCNC-T P Brandler
Natick MA 01760-5002

Director
US Army Rsrch Ofc
4300 S Miami Blvd
Research Triangle Park NC 27709

US Army Simulation, Train, & Instrmntn Cmnd
Attn J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army Tank-Automtv Cmnd Rsrch, Dev, & Engrg Ctr
Attn AMSTA-TA J Chapin
Warren MI 48397-5000

US Army Train & Doctrine Cmnd Battle Lab Integration & Techl Dirctrt
Attn ATCD-B J A Klevecz
FT Monroe VA 23651-5850

US Military Academy
Mathematical Sci Ctr of Excellence
Attn MDN-A MAJ M D Phillips
Dept of Mathematical Sci Thayer Hall
West Point NY 10996-1786

Nav Surface Warfare Ctr
Attn Code B07 J Pennella
17320 Dahlgren Rd Bldg 1470 Rm 1101
Dahlgren VA 22448-5100

DARPA
Attn B Kaspar
3701 N Fairfax Dr
Arlington VA 22203-1714

Elect Commctn Lab
Attn J L Kurtz
PO Box 140245
Gainesville FL 32608

Hicks & Associates, Inc
Attn G Singley III
1710 Goodrich Dr Ste 1300
McLean VA 22102

Distribution (cont'd)

Palisades Inst for Rsrch Svc Inc
Attn E Carr
1745 Jefferson Davis Hwy Ste 500
Arlington VA 22202-3402

US Army Rsrch Lab
Attn AMSRL-D R W Whalin
Attn AMSRL-DD J Rocchio
Attn AMSRL-CI-LL Techl Lib (3 copies)
Attn AMSRL-CS-AS Mail & Records Mgmt
Attn AMSRL-CS-EA-TP Techl Pub (3 copies)

US Army Rsrch Lab (cont'd)
Attn AMSRL-SE J Pellegrino
Attn AMSRL-SE-RM E Adler
Attn AMSRL-SE-RM G Goldman
Attn AMSRL-SE-RM R Wellman
Attn AMSRL-SE-RU B Scheiner (10 copies)
Attn AMSRL-SE-RU L Nguyen
Attn AMSRL-SE-RU R Innocenti
Attn AMSRL-SE-RU T Ton
Adelphi MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1999	Final 7/98 to 9/98

4. TITLE AND SUBTITLE A MATLAB Radar Range Equation and Probability of Detection Evaluation Tool

5. FUNDING NUMBERS
DA PR: AH94
PE: 62705A

6. AUTHOR(S) Barry Scheiner

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL-SE-RU email: bscheiner@arl.mil
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1806

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
ARL PR: 9NE6l1
AMS code: 622705.H94

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

I have developed a graphical user interface (GUI) that makes the process of evaluating the radar range equation faster and more convenient. From this GUI, one can enter the input parameters necessary to compute the signal-to-noise ratio (SNR) and probability of detection (P_d) of a radar system as a function of range, and view the results graphically. The GUI and the SNR and P_d computations are coded using the MATLAB programming language. I review the derivation of the equation used to evaluate the P_d of a signal detected by a linear quadrature detector and provide examples of the GUI's operation.

14. SUBJECT TERMS
GUI, graphical user interface

15. NUMBER OF PAGES
24

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102